Tecnologie Web T (9 cfu)
Prova d'Esame – 09 Giugno 2022 – Versione A

Tempo a disposizione: 180 minuti

La soluzione comprende la consegna elettronica dei seguenti file:

A1.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
A2.zip	file zip contenente il sorgente java/class e pagine Web per punto 2
A3.zip	file zip contenente pagine Web e codice React.js per punto 3

Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto.

N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), ben distribuiti sui 3 esercizi, ovvero in ciascuno dei tre esercizi si deve raggiungere una valutazione almeno quasi sufficiente.

ESERCIZIO 1 (12 punti)

Si realizzi una applicazione Web per **l'elaborazione casuale server-side di un testo**, basandosi principalmente sulle tecnologie Java Servlet e JSP.

L'applicazione Web deve permettere a utenti non autenticati di inserire un testo di lunghezza compresa fra 100 e 1000 caratteri; si controlli che tale testo contenga solo caratteri alfabetici e numerici. Una volta inseriti almeno 100 caratteri validi, deve attivarsi un bottone per l'invio dei dati al servitore; alla pressione del bottone il testo inserito deve essere passato al server in formato JSON. Lato servitore una **servlet S1** dovrà occuparsi di estrarre a sorte un numero naturale *i* compreso fra 0 e 9, e di eliminare dal testo ricevuto tutti i caratteri di posizione multipla di *i*; il risultato dell'elaborazione di S1 dovrà essere inviato a una **JSP J2**, che dovrà contare il numero di maiuscole presenti nel testo risultante. Il testo modificato più il conteggio dovranno essere restituiti al cliente, anche in questo caso in formato JSON.
Si faccia inoltre in modo che, nel caso di arrivo di una richiesta cliente a S1 mentre ci sono ancora almeno 3 richieste di servizio in esecuzione lato server, questa richiesta venga **"bloccata" per 10 secondi**, mostrando nel frattempo una pagina Web di risposta con la scritta "Server sovraccarico: la richiesta viene rallentata di 10 secondi".

In ogni momento di esecuzione dell'applicazione, previa autenticazione, un amministratore deve avere la possibilità di visualizzare il numero di richieste totali di servizio sottomesse da utenti non amministratori a partire dall'inizio dell'esecuzione dell'applicazione.

Tecnologie Web T (9 cfu)
Prova d'Esame – 09 Giugno 2022 – Versione A

ESERCIZIO 2 (10 punti)

Si realizzi una applicazione Web per **il calcolo parallelo degli anagrammi di una parola di al max 8 caratteri**. L'applicazione Web deve essere basata principalmente su tecnologie Javascript, AJAX e servlet.

In particolare, l'applicazione Web deve permettere a utenti non autenticati di inserire una parola di al max 8 caratteri, controllando che i caratteri inseriti siano tutti alfabetici minuscoli, terminata dal carattere speciale "$". All'inserimento di $, l'applicazione deve invocare automaticamente l'esecuzione parallela server-side del calcolo dei possibili anagrammi da parte di 2 istanze concorrenti di **servlet**, **S1** e **S2**. S1 dovrà calcolare 10 possibili anagrammi che cominciano per vocale, generati in modo casuale riordinando i caratteri della stringa di input; similmente, S2 calcolerà 10 possibili anagrammi che cominciano per consonante. Terminato il calcolo del decimo anagramma, ciascuna delle servlet dovrà inviare immediatamente le 10 stringhe al cliente, dove l'utente potrà segnalare se una delle stringhe inviate da S1 o da S2 è di senso compiuto in lingua italiana; la servlet corrispondente risulterà la vincitrice del gioco.

Una volta che l'utente ha indicato la servlet vincitrice, l'esecuzione dell'altra servlet concorrente non è più necessaria e ne deve essere forzata la terminazione il prima possibile.

ESERCIZIO 3 (11 punti)

Si realizzi in React un'applicazione Web lato client per il gioco del "Lotto semplificato". L'applicazione dovrà eseguire interamente sul browser senza interagire con alcun server remoto.
L'interfaccia dell'applicazione sarà composta da tre sezioni:

- Una sezione che espone funzionalità per la **compilazione della schedina** (sezione "compilazione"). Una schedina è composta da 5 campi di testo per l'immissione di altrettanti numeri che possono assumere un valore da 1 a 10 (numeri "ammissibili"). L'utente avrà la possibilità di generare una schedina vuota attraverso un bottone, compilare i cinque campi di testo della schedina e finalizzare la giocata attraverso un altro bottone. La finalizzazione, previo controllo che ogni casella contenga un numero ammissibile, produrrà una schedina valida da visualizzare nella sezione "visualizzazione";
- Una sezione per la **visualizzazione della schedina** compilata (sezione "visualizzazione"). In questa sezione verranno inizialmente visualizzati cinque campi di testo, a sola lettura, contenenti la sequenza dei numeri giocati;
- Una sezione per l'**estrazione dei numeri** (sezione "estrazione"). Questa sezione conterrà un bottone per avviare l'estrazione casuale di cinque numeri ammissibili (tutti diversi tra loro) e cinque campi di testo a sola lettura in cui appariranno tali numeri. Ad estrazione avvenuta, la schedina compilata dall'utente potrà risultare vincente (se almeno uno fra gli eventi "ambo", "terno", "quaterna" o "cinquina" si sarà verificato) oppure non-vincente. Occorrerà aggiornare la sezione visualizzazione evidenziando questa informazione in testa alla sequenza dei numeri. Inoltre, sempre nella sezione visualizzazione: nel caso di "ambo", lo sfondo dei due campi di testo contenenti i numeri dell'ambo si colorerà di giallo; nel caso di terno, lo sfondo dei tre campi di testo contenenti i numeri del terno si colorerà di verde; nel caso di quaterna di blu; nel caso di cinquina di rosso.